In the Claims:

The following is a current claims listing for the present application.

1. (Canceled)

2. (Previously presented) The system of claim 32, wherein the cache accumulator memory comprises a dual-ported memory.

3. (Previously presented) The system of claim 32, wherein the cache accumulator memory comprises at least two independently interfaced memory banks, wherein the cache accumulator memory is configured to provide the block operand from a first one of the independently interfaced memory banks and to store the block result in a second one of the independently interfaced memory banks.

4. (Previously presented) The system of claim 32, wherein the cache accumulator memory is configured to indicate whether a particular block operand stored in the cache accumulator memory is modified with respect to a copy of that particular block operand in the memory.

5. (Previously presented) The system of claim 32, wherein the cache accumulator memory is configured to load a copy of the block operand into the cache accumulator memory from the memory in response to the block operand not being present in the cache accumulator memory when the instruction is received.

6. (Previously presented) The system of claim 5, wherein the cache accumulator memory comprises a plurality of block storage locations, wherein if all of the block storage locations are currently storing valid data, the cache accumulator memory is configured to select one of the block storage locations to overwrite with the copy of the block operand and to load the copy of the block operand into the selected one of the block storage locations.

7. (Previously presented) The system of claim 6, wherein the cache accumulator memory is configured to use a least recently used algorithm to select the one of the block storage locations to overwrite.

8. (Previously presented) The system of claim 6, wherein if data in the selected one of the block storage locations is modified with respect to a copy of that data in the memory, the cache accumulator memory is configured to write the data back to the memory before loading the copy of the block operand into the selected one of the block storage locations.

9. (Previously presented) The system of claim 32, wherein the functional unit is configured to perform a parity calculation on the block operand.

10. (Canceled)

11. (Previously presented) The system of claim 32, wherein the functional unit is configured to calculate a parity block from a plurality of data blocks in a stripe of data, wherein the first block operand is a first one of the data blocks in the stripe of data.

12. (Previously presented) The system of claim 32, wherein the functional unit is configured to perform the block operation on two block-operands.

13. (Previously presented) The system of claim 12, wherein a first of the two block-operands is the first block operand stored in the cache accumulator memory and a second of the two block-operands is provided on a data bus coupled to provide operands to the functional unit.

14. (Previously presented) The system of claim 12, wherein a first of the two block-operands is the first block operand stored in the cache accumulator memory and a second of the two block-operands is provided from the memory to the functional unit.

15. (Previously presented) The system of claim 32, wherein the cache accumulator memory is configured to provide a word of the first block operand to the functional unit during an access cycle in which the cache accumulator memory also stores a word of the block result generated by the functional unit.

16. (Canceled)

17. (Previously presented) The method of claim 33, wherein said providing comprises providing successive words of the first block operand and wherein said accumulating comprises storing successive words of the block result, wherein a word of the first block operand is provided from the cache accumulator memory to the functional unit during an access cycle in which a word of the block result is stored in the cache accumulator memory.

18. (Previously presented) The method of claim 33, wherein the cache accumulator memory comprises a dual-ported memory, wherein said accumulating comprises overwriting the first block operand with the block result.

19. (Previously presented) The method of claim 33, wherein the cache accumulator memory comprises at least two independently interfaced memory banks, wherein said loading comprises loading the first block operand into a first one of the independently interfaced memory banks and wherein said accumulating comprises storing the block result in a second one of the independently interfaced memory banks.

20. (Previously presented) The method of claim 33, wherein the cache accumulator memory comprises a plurality of block storage locations, wherein if all of the block storage locations are currently storing valid data when the first command is received, said loading comprises selecting one of the block storage locations to overwrite with the copy of the first block operand and loading the copy of the first block operand into the selected one of the block storage locations.

21. (Original) The method of claim 20, wherein said selecting comprises using a least recently used algorithm to select the one of the block storage locations to overwrite.

22. (Original) The method of claim 20, further comprising writing data in the selected one of the block storage locations back to the memory if the data is modified with respect to a copy of that data in the memory.

23. (Previously presented) The method of claim 33, further comprising the functional unit performing a parity calculation on the first block operand to generate the block result in response to said providing.

24. (Previously presented) The method of claim 33, wherein the operation comprises a parity calculation, and wherein the command is issued by an array controller configured to perform block operations on data stored to the storage array.

25. (Previously presented) The method of claim 33, further comprising the functional unit performing the operation on the first block operand and a second block operand in response to said providing.

26. (Original) The method of claim 25, further comprising a data bus providing the second block operand to the functional unit.

27. (Canceled)

28. (Previously presented) A data processing system, comprising:

a host computer system;

a storage array;

an interconnect coupled to the host computer system and the storage array and configured to transfer data between the host computer system and the storage array; and

a parity calculation system configured to perform parity operations on data stored to the storage array, wherein the parity calculation system comprises a memory configured to provide an addressable block operand storage space and to store within the block operand storage space block operands received from the storage array, a cache accumulator memory comprising a plurality of block storage locations, and a parity calculation unit;

wherein in response to an instruction using an address in the memory to identify the first block operand, the cache accumulator memory is configured to:

determine whether the first block operand is stored among the plurality of block storage locations of the cache accumulator memory;

in response to determining that the first block operand is stored among the plurality of block storage locations of the cache accumulator memory, output the first block operand from the plurality of block storage locations to the parity calculation unit and subsequently store a first block result generated by the parity calculation unit within the plurality of block storage locations of the cache accumulator memory;

in response to determining that the first block operand is not stored among the plurality of block storage locations of the cache accumulator memory, initiate a fetch operation to fetch the first block operand from the memory, and upon receiving the first block operand from the memory, store the first block operand within the plurality of block storage locations of the cache accumulator memory;

wherein the plurality of block storage locations of the cache accumulator memory are configured to cache a portion of the block operand storage space of the memory and to accumulate an intermediate result of a block accumulation operation performed on a given block operand, wherein the intermediate result is both a result of and an operand of the block accumulation operation, such that during the block accumulation operation, the plurality of block storage locations of the cache accumulator memory are concurrently configured both to cache certain ones of the block operands and to accumulate the intermediate result of the block accumulation operation.

29. (Previously presented) The data processing system of claim 28, wherein the parity calculation unit is configured to perform a parity calculation on the first block operand provided by the cache accumulator memory and a second block operand provided on a data bus.

30. (Original) The data processing system of claim 29, wherein the parity calculation system is configured to calculate a parity block from a plurality of data blocks in a stripe of data, wherein the first block operand is a first one of the data blocks in the stripe of data and wherein the second block operand is a second one of the data blocks in the stripe of data.

31. (Previously presented) The data processing system of claim 28, wherein the cache accumulator memory is configured to store a word of the first block result during an access cycle in which the cache accumulator memory provides a word of the first block operand to the parity calculation unit.

32. (Previously presented) A system, comprising:

a storage array including a plurality of mass storage devices;

an array controller configured to perform block operations on data stored to the storage array, wherein the array controller includes a memory configured to provide an addressable block operand storage space and to store within the block operand storage space block operands received from one or more of said plurality of mass storage devices; and a cache accumulator memory comprising a plurality of block storage locations and a functional unit configured to perform a block operation on one or more block operands to generate a block result;

wherein the plurality of block storage locations of the cache accumulator memory are configured to cache a portion of the block operand storage space of the memory;

wherein in response to an instruction using an address in the memory to identify a first block operand, the cache accumulator memory is configured to:

determine whether the first block operand is stored among the plurality of block storage locations of the cache accumulator memory;

in response to determining that the first block operand is stored among the plurality of block storage locations of the cache accumulator memory, output the first block operand from the plurality of block storage locations to the functional unit; and

in response to determining that the first block operand is not stored among the plurality of block storage locations of the cache accumulator memory, initiate a fetch operation to fetch the first block operand from the memory, and upon receiving the first block operand from the memory, store the first block operand within the plurality of block storage locations of the cache accumulator memory;

wherein the plurality of block storage locations of the cache accumulator memory are further configured to accumulate an intermediate result of a block accumulation operation performed on the first block operand, wherein the intermediate result is both a result of and an operand of the block accumulation operation, such that during the block accumulation operation, the plurality of block storage locations of the cache accumulator memory are concurrently configured both to cache certain ones of the block operands and to accumulate the intermediate result of the block accumulation operation.

33. (Previously presented) A method of performing a block accumulation operation, the method comprising:

storing data to a storage array including a plurality of mass storage devices;

receiving a first command to perform a block accumulation operation on a first block operand identified by a first address in a memory, wherein the first block operand corresponds to data stored to the storage array, and wherein the memory is configured to provide an addressable block operand storage space and to store within the block operand storage space block operands received from one or more of said plurality of mass storage devices;

in response to receiving the first command:

determining whether the first block operand is stored among a plurality of block storage locations of a cache accumulator memory, wherein the plurality of block storage locations are configured to cache a portion of the block operand storage space of the memory;

loading the first block operand from the memory into one of the plurality of block storage locations included in the cache accumulator memory ~~if~~ in response to determining that the first block operand is not stored in the cache accumulator memory;

in response to determining that the first block operand is stored in the cache accumulator memory, providing the first block operand from the plurality of block storage locations of the cache accumulator memory to a functional unit; and

accumulating a block result of the block accumulation operation generated by the functional unit into the plurality of block storage locations, wherein the block result is both a result of and an operand of the block accumulation operation, such that during the block accumulation operation, the plurality of block storage locations of the cache accumulator memory are concurrently configured both to cache certain ones of the block operands and to accumulate the intermediate result of the block accumulation operation.

34. (Previously presented) An apparatus, comprising:

storage array means configured for storing data; and

means for performing block operations on data stored to the storage array means, wherein the means for performing block operations is configured to generate block results; and

means for accumulating block results generated by the means for performing block operations, wherein the means for accumulating block results is coupled to the means for performing block operations and comprises a plurality of block storage locations configured to cache a portion of an addressable block operand storage space of a memory configured to store within the block operand storage space block operands received from the storage array means;

wherein in response to an instruction that uses an address in the memory to identify a first block operand within the addressable block operand storage space, the means for accumulating block results is further configured to:

determine whether the first block operand is stored among the plurality of block storage locations of the means for accumulating block results;

in response to determining that the first block operand is stored among the plurality of block storage locations of the means for accumulating block results, provide the first block operand to the means for performing a first block operation;

in response to determining that the first block operand is not stored among the plurality of block storage locations of the means for accumulating block results, initiate a fetch operation to fetch the first block operand from the memory, and upon receiving the first block operand from the memory, store the first block operand within the plurality of block storage locations of the means for accumulating block results;

wherein the plurality of block storage locations of the means for accumulating block results are configured to accumulate an intermediate result of a block accumulation operation performed on the first block operand, wherein the intermediate result is both a result of and an operand of the block accumulation operation, such that during the block accumulation operation, the plurality of block storage locations of the means for accumulating block results are concurrently configured both to cache certain ones of the block operands and to accumulate the intermediate result of the block accumulation operation.